



**Università degli Studi di Salerno
Anno Accademico 2018/2019**

Corso di Ingegneria del Software

System Design Document

V1.7



STAY SOFTWARE

Top Manager:

Prof. De Lucia Andrea

Team di sviluppo:

Nome e Cognome	Matricola
<i>Aniello Mancusi</i>	0512102610
<i>Vincenzo Zito</i>	0512100507

Revision History:

Autore	Data	Descrizione	Versione
<i>Aniello Mancusi</i>	<i>27/02/19</i>	<i>Struttura documento</i>	<i>v 1.0</i>
<i>Vincenzo Zito</i>	<i>27/02/19</i>	<i>Stesura generale del documento</i>	<i>v 1.0</i>
<i>Vincenzo Zito</i>	<i>27/02/19</i>	<i>Stesura capitolo "Architettura del sistema proposto"</i>	<i>v 1.1</i>
<i>Aniello Mancusi</i>	<i>28/02/19</i>	<i>Stesura capitolo "Gestione del controllo globale"</i>	<i>v 1.2</i>
<i>Aniello Mancusi</i>	<i>28/02/19</i>	<i>View sottosistemi</i>	<i>v 1.3</i>
<i>Vincenzo Zito</i>	<i>01/03/19</i>	<i>Stesura capitolo "Gestione dei dati persistenti"</i>	<i>v 1.4</i>
<i>Vincenzo Zito</i>	<i>01/03/19</i>	<i>Revisione capitolo 1, 2, 3.1, 3.2, 3.3, 3.5, 3.6, 3.7, 4</i>	<i>v 1.5</i>
<i>Aniello Mancusi</i>	<i>01/03/19</i>	<i>Revisione totale documento</i>	<i>v 1.6</i>
<i>Aniello Mancusi</i>	<i>23/10/2019</i>	<i>Aggiunta diagrammi design time</i>	<i>v1.7</i>

Sommario

Introduzione	4
Scopo del sistema	4
Design goals.....	4
Criteri di prestazione.....	4
Criteri di affidabilità	5
Criteri di manutenzione	6
Definizioni, acronimi e abbreviazioni	7
Overview	7
Architettura del sistema	8
Architettura del sistema proposto	8
Overview	8
Decomposizione in sottosistemi	8
Servizi sottosistemi	9
Interface layer	9
Application Logic Layer	9
Storage layer.....	9
View sottosistemi	10
Hardware/Software mapping	10
Specifiche tecniche mapping hardware.....	11
Terminale Utente	11
Server Gestore.....	11
Server Database	12
Diagrammi Design Time	12
Visualizza.....	12
Aggiunta	13
Modifica	13
Riparazione	14
Gestione dei dati persistenti.....	14
Schema EER.....	14
Dizionario dei dati.....	15
Relazioni	15
Contiene:	15
Effettua:	15
Richiede:	15
Ordine:	15
Amministratore:	15
Cliente:.....	16
Riparazione:	16
Prodotto:	16
Codice di creazione del database	17
Controllo degli accessi e sicurezza.....	19
Controllo degli accessi	19
Sicurezza	20
Gestione del controllo globale	20
Boundary conditions	20
Configuration	20
Exception handling.....	21
Glossario.....	23

Introduzione

Scopo del sistema

Il sistema, automatizzando la maggior parte delle operazioni che si svolgono in un normale negozio di informatica, renderà più fluido e leggero il carico di lavoro di chi è addetto all'amministrazione del suddetto negozio. Le operazioni di carico merci, di acquisto/vendita prodotti e le riparazioni saranno gestite dal sistema che si verrà a creare e quindi i dati verranno conservati in un database relazionale creato ad-hoc e la gestione sarà affidata ad una web application disegnata su misura per il cliente.

Design goals

Il sistema sarà progettato con un'interfaccia user-friendly in modo da agevolare l'utilizzo a persone che non hanno molta dimestichezza con le web-application di questo genere. Il software sarà ideato in modo da gestire al meglio le risorse del sistema evitando fastidiosi "crash". Sarà un prodotto utile e semplice nell'interazione che permetterà ai propri utenti di trovarsi davanti un software facile da utilizzare. La gestione del database e le relative operazioni saranno completamente gestibili dall'interfaccia di gestione web che sarà creata ad-hoc secondo le richieste del cliente e attraverso l'invio di numerosi feedback per ogni operazione il proprietario sarà agevolato nel cercare di commettere meno errori possibili. È prevista la realizzazione di una web application accessibile facilmente anche da dispositivi mobile, dato che sono i più utilizzati, attraverso la costruzione di un'interfaccia responsive. Dal lato client l'interfaccia sarà il più semplice possibile per permettere una ricerca veloce ed intuitiva dei prodotti e per agevolare l'acquisto di un bene/servizio in maniera semplice e con una serie finita di passi.

Criteri di performance

<i>Tempo di risposta</i>	Il sistema deve essere efficiente, deve garantire una risposta in breve tempo, massimo 10 secondi, senza che si verifichino errori.
<i>Throughput</i>	Il sistema riesce a soddisfare massimo 1000 task contemporaneamente in un tempo prefissato di un minuto.
<i>Memoria</i>	La memoria verrà gestita dal sistema il quale oltre a gestirne la memoria fisica (RAM), gestisce anche la memoria secondaria (Hard Disk).

Criteri di affidabilità

<i>Robustezza</i>	Il sistema deve gestire eventuali input errati da parte dell'utente, attivando politiche di controllo sul lato server, utilizzando PHP. Il tutto deve avvenire senza interrompere il funzionamento dell'intero sistema.
<i>Sicurezza</i>	Il sistema prevede il controllo degli accessi e dell'autenticazione, permettendo l'uso delle varie funzionalità alle categorie di utente specificate nella tabella degli accessi
<i>Tolleranza crash</i>	In caso di errore il sistema deve continuare ad operare come se non ci fossero problemi, in modo trasparente all'utente. Al verificarsi di un crash il sistema salverà le impostazioni utili nel database tralasciando informazioni non corrette o non complete.
<i>Affidabilità</i>	Il sistema oltre alla gestione degli errori deve garantire il funzionamento 24h su 24h tranne nei tempi relativi alla manutenzione ordinaria che da contratto prevede un tempo massimo di stop di 12h.

Criteri di Costo

<i>Sviluppo</i>	I costi di sviluppo previsti riguardano l'affitto del server per la fase di sviluppo. Per quanto riguarda il DBMS utilizzeremo MySQL che è open source. Inoltre i costi di sviluppo prevedono la retribuzione per il lavoro dei singoli componenti del team. Supponendo di quantificare tale retribuzione stimiamo 5000€ a persona a prescindere dal ruolo ricoperto, per un totale di (35000€). In aggiunta sono previsti (919€ + 799€ + 800€ + 700€) di licenze software: Balsamiq per 12 persone, Visual Paradigm Professional , PhotoShop, Microsoft Office, MySQL.
<i>Costo amministrazione</i>	Per i costi di amministrazione si stima il pagamento annuo di circa 1200 €.

Criteri di manutenzione

<i>Estendibilità</i>	<p>Ai fini di uno sviluppo futuro deve essere possibile aggiungere, in un secondo momento, altre funzionalità al sistema.</p> <p>Per rendere possibile questo, la piattaforma verrà programmata nel modo più semplice possibile cercando di dividere al meglio le funzionalità in modo da poter, in futuro, apportare modifiche anche a singole parti e non per forza stravolgendo l'intero sistema.</p>
<i>Leggibilità</i>	<p>Il codice deve essere chiaro, quindi i nomi delle variabili e dei metodi che verranno utilizzati devono essere coerenti al contesto d'uso, dove è strutturato, per rendere più veloce una modifica o una estensione del sistema anche da parte di terzi che si avvicinano al progetto in un secondo momento.</p>
<i>Modificabilità</i>	<p>Deve essere possibile intervenire sul codice esistente per correggere eventuali bugs o implementare nuove funzionalità. Bisogna garantire che il codice sia leggibile per rendere agevole la modifica.</p>
<i>Tracciabilità dei requisiti</i>	<p>Grazie alla tracciabilità dei requisiti, sarà possibile effettuare le modifiche necessarie al corretto funzionamento del sistema, valutando correttamente i costi e i rischi che le modifiche porteranno.</p>
<i>Portabilità</i>	<p>La portabilità sarà garantita dall'utilizzo di un web container quale Apache Tomcat. L'applicazione, infatti, una volta compilata, potrà essere eseguita su una qualsiasi server Tomcat compatibile con la versione per cui è stato compilato.</p>

Criteri per l'utente finale

<i>Usabilità</i>	<p><i>StaySoftware</i> deve essere facilmente apprendibile (l'utente deve essere in grado di interagire con il sistema e padroneggiare le funzionalità in modo rapido), deve essere flessibile e robusto (l'utente deve essere in grado di capire quando ha successo nel perseguire i suoi obiettivi nel sistema o quando sta sbagliando qualcosa per poter intervenire).</p>
-------------------------	---

Definizioni, acronimi e abbreviazioni

- StaySoftware: Nome del sistema che verrà sviluppato.
- RAD: Requirement Analysis Document
- DBMS: Database Management System

Overview

Il presente documento SDD (System Design Document) si occupa di:

- 1) Definire gli obiettivi di design del progetto (identificando quali caratteristiche di qualità dovrebbero essere ottimizzate);
- 2) Decomporre il sistema in vari sottosistemi più piccoli che sono di più facile realizzazione;
- 3) Selezionare le strategie per costruire il sistema:
 - a) Strategia Hardware/Software;
 - b) Strategie che si riferiscono alla gestione dei dati persistenti;
 - c) Il flusso di controllo globale;
 - d) Le politiche di controllo degli accessi;
 - e) La gestione delle condizioni limite;

L'output del system design è un modello del sistema che include quindi la decomposizione dello stesso in sottosistemi con la relativa descrizione per ognuna delle strategie utilizzate.

Architettura del sistema

L'architettura attualmente proposta non va a sostituire nessuna architettura esistente, il sistema è alla sua prima versione.

Architettura del sistema proposto

Overview

Il software è composto seguendo lo schema architetturale client/server

- **Client:** il browser preferito del cliente.
- **Server:** le pagine PHP che implementano l'intero sistema.

Sono possibili tre configurazioni:

- **Single-Tier:** tutto il sistema è distribuito su una sola macchina che funzionerà da client e da server.
- **Two-Tier:** il sistema è distribuito ed eseguito su due macchine differenti; una macchina gestirà tutto il lato server e conterrà anche il database mentre l'altra macchina eseguirà il client.
- **Three-Tier:** a differenza della configurazione Two-Tier il database verrà gestito su una macchina separata.

La configurazione scelta è quella three-tier, così eventuali modifiche ad uno dei tre moduli non comporterà il blocco del sistema e la relativa modifica dei blocchi rimanenti rispetto a quello aggiornato. Tale configurazione è basata su tre layer:

- 1) **Interface Layer:** livello che gestisce la parte grafica del sistema software
- 2) **Application Logic Layer:** gestisce la parte logica e le relative query che vengono sottomesse allo storage layer.
- 3) **Storage Layer:** questo livello gestisce l'archiviazione persistente dei dati.

Decomposizione in sottosistemi

I layer previsti sono: *Interface Layer*, *Application Logic Layer*, *Storage Layer*. Di seguito riportiamo, per ogni modulo, le componenti da cui è composto, che rappresentano nel dettaglio le operazioni di cui si occupa e i dati su cui operano.

Servizi sottosistemi

Di seguito riportiamo i servizi forniti da ognuno dei sottosistemi in termini di operazioni

Interface layer

Nome	<i>Interface layer</i>
Descrizione	Offre all'utente servizi di interazione uomo-macchina, ovvero click, submit etc.
Moduli presenti	
Modulo	Descrizione
<i>Interfaccia web</i>	Questo modulo descrive l'interfaccia grafica con cui l'utente interagisce e che interpreta i form sottomessi e quindi di ricevere e interpretare gli input.

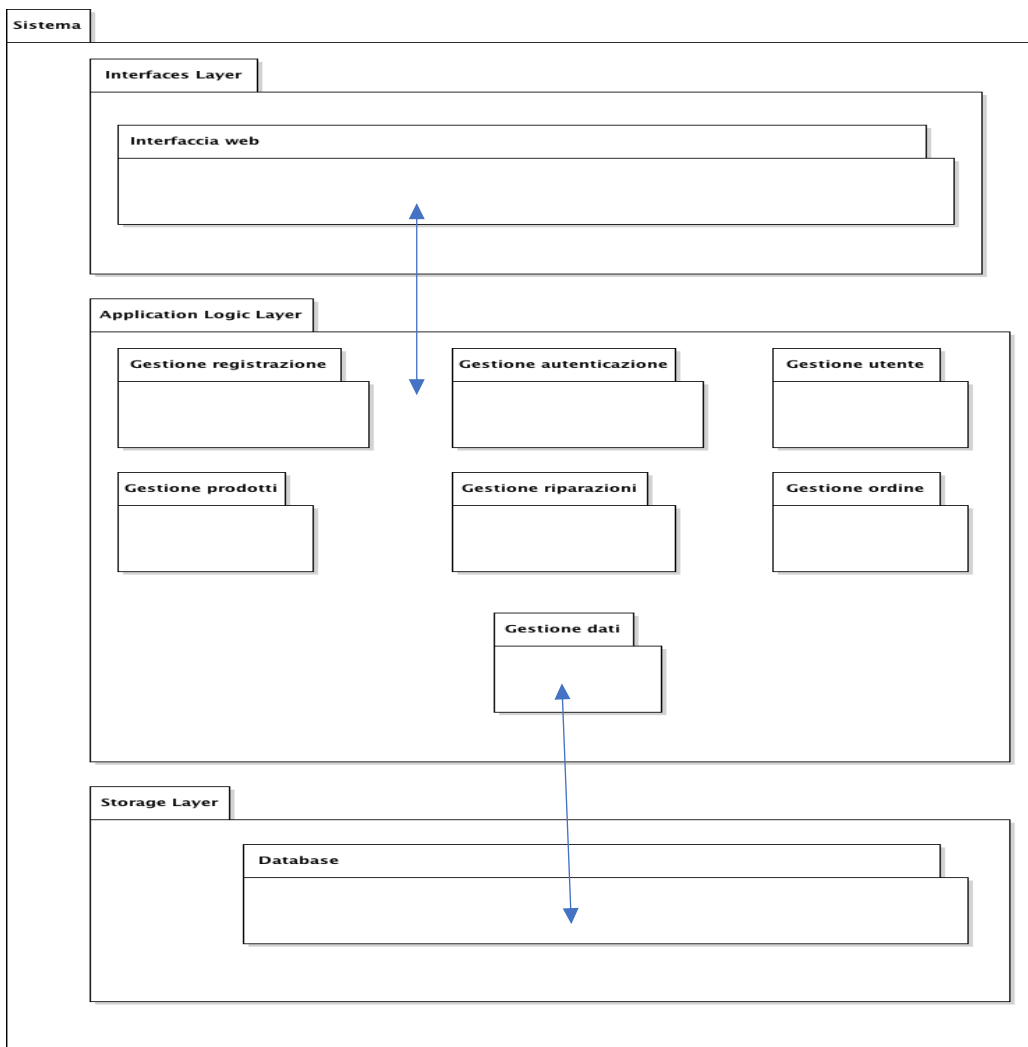
Application Logic Layer

Nome	<i>Application Logic Layer</i>
Descrizione	Questo livello gestisce la parte logica e le relative query.
Modulo	Descrizione
<i>Gestione registrazione</i>	Il modulo consente la registrazione di un cliente sul database quindi la verifica dei dati, la presenza di un cliente con dati uguali e la memorizzazione nel database.
<i>Gestione autenticazione</i>	Il modulo consente il login e il logout e le relative funzioni di esecuzione e controllo.
<i>Gestione utente</i>	Il modulo si occupa della gestione generale dell'utente dei dati di registrazione, nome, cognome, telefono, indirizzo ed email.
<i>Gestione prodotti</i>	Il modulo gestisce i prodotti e racchiude nello specifico le funzionalità ad esso relative (inserimento, modifica).
<i>Gestione riparazioni</i>	Il modulo gestisce le riparazioni e quindi i relativi cambi di stato che comportano modifiche anche a livello di interfaccia grafica.
<i>Gestione ordine</i>	Il modulo gestisce gli ordini
<i>Gestione dati</i>	Il modulo si occupa della comunicazione con il database

Storage layer

Nome	<i>Storage layer</i>
Descrizione	Questo livello gestisce l'archiviazione persistente dei dati.
Moduli presenti	
Modulo	Descrizione
<i>Database</i>	Si occupa del database e della sua gestione.

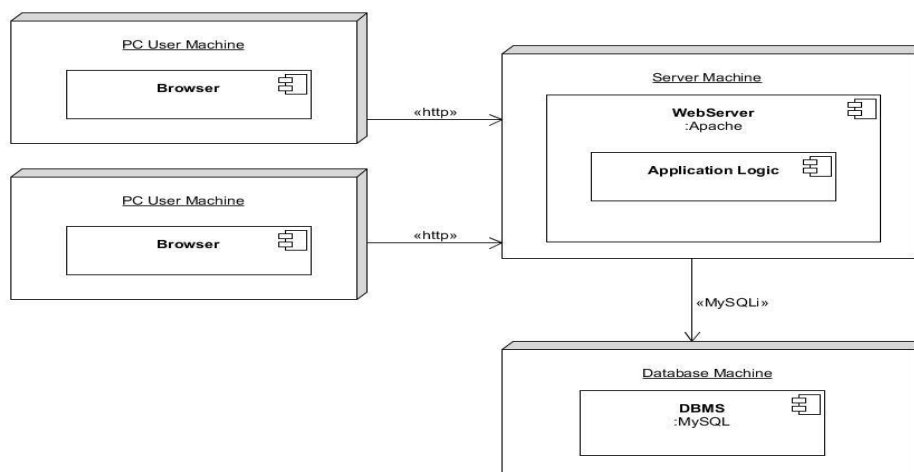
View sottosistemi



Hardware/Software mapping

Nell'applicazione proposta i Client sono Web browser, mentre il Web Server contiene le componenti che si occupano della logica applicativa. Per realizzare le componenti del Client vengono utilizzate pagine php che permettono di creare gli oggetti Boundary.

Il Web server invia query al Database MySQL utilizzando il protocollo MySQLi.



Specifiche tecniche mapping hardware

Web Server: Apache versione 2.4.38

DBMS: MySQL versione 8.0.15

Il software utilizzerà un'architettura Three-Tier in configurazione Client-Server-Database:

- **Client:** si intende la macchina utilizzata dall'utente e dove gira il browser che quest'ultimo utilizza per navigare sul sito.
- **Server:** si intende la macchina su cui gira l'applicazione: le pagine php, le query.
- **Database:** si intende la macchina su cui il database è installato e che gestisce quindi le richieste di dati sia in input che in output.

Terminale Utente

L'utente (inteso genericamente) accede al sistema attraverso il tuo terminale (primo nodo). Sul client non risiederà la parte principale del sistema, ma in base alle scelte dell'utente ed a login e password inserite, il terminale visualizzerà una schermata diversa, a seconda che si tratti di un utente registrato o di un amministratore. La sua funzionalità, quindi, è quella di visualizzare all'utente le pagine del sistema (ignorando che siano statiche o dinamiche), caricando le informazioni grazie ad una connessione al sito.

Requisiti minimi per il nodo "Terminale utente":

- | | |
|-----------------|----------|
| ○ Processore: | 1600Mhz |
| ○ RAM: | 256MB |
| ○ Disco rigido: | 1GB |
| ○ Connessione: | LAN -100 |

Server Gestore

Il secondo nodo è quello dove risiederà il web-server, contenente tutte le informazioni relative all'intero dominio di applicazione. Su tale macchina devono essere installati alcuni software necessari ai servizi offerti dal sistema, come Apache e il modulo per gestire PHP, indispensabile per la gestione della comunicazione con il database e la generazione delle pagine da far visualizzare al client.

Requisiti minimi per il nodo "Server Gestore":

- | | |
|-----------------|---------|
| ○ Processore: | 3.2 Ghz |
| ○ RAM: | 2GB |
| ○ Disco rigido: | 500GB |
| ○ Connessione: | LAN-100 |

Server Database

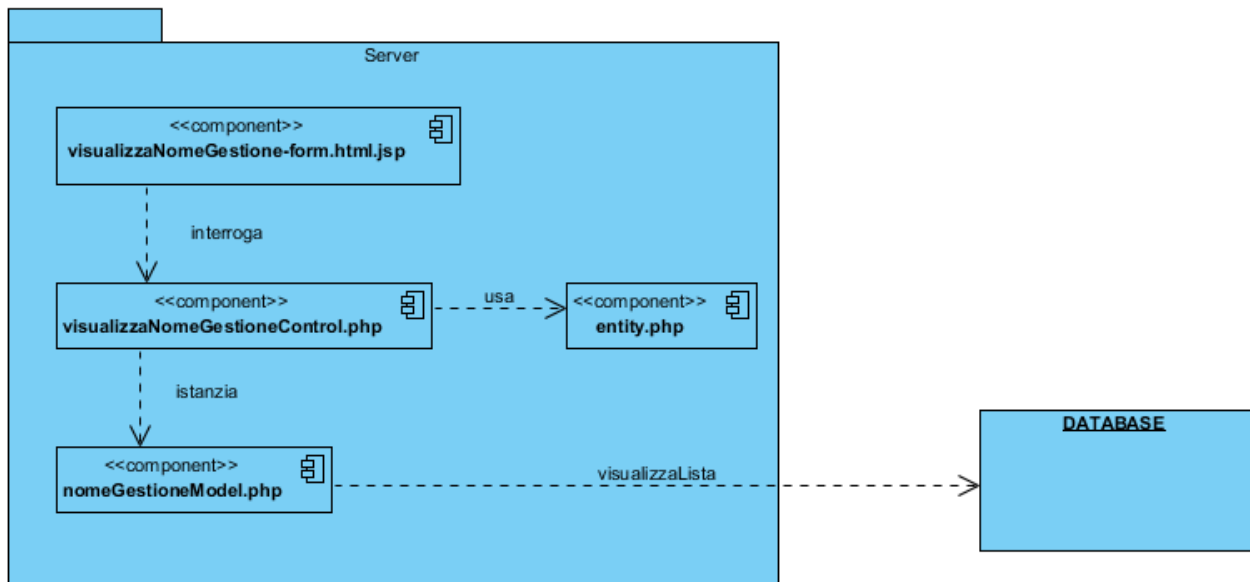
Il terzo nodo è quello dove risiederà il database fisico, contenete tutte le informazioni relative all'intero dominio di applicazione. Su tale macchina devono essere installati alcuni software necessari ai servizi offerti dal sistema, come il DBMS MySQL, indispensabile per la gestione dei dati persistenti.

Requisiti minimi per il nodo “Server Database”:

- Processore: 3.2 Ghz
- RAM: 2GB
- Disco rigido: 500GB
- Connessione: LAN-100

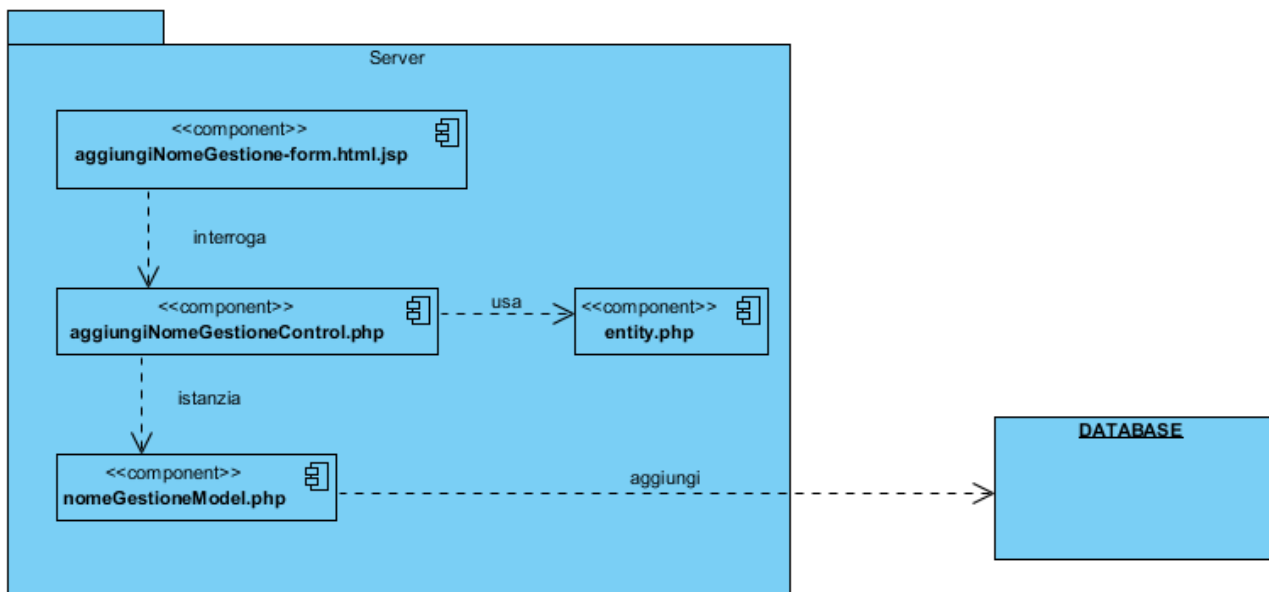
Diagrammi Design Time

Visualizza



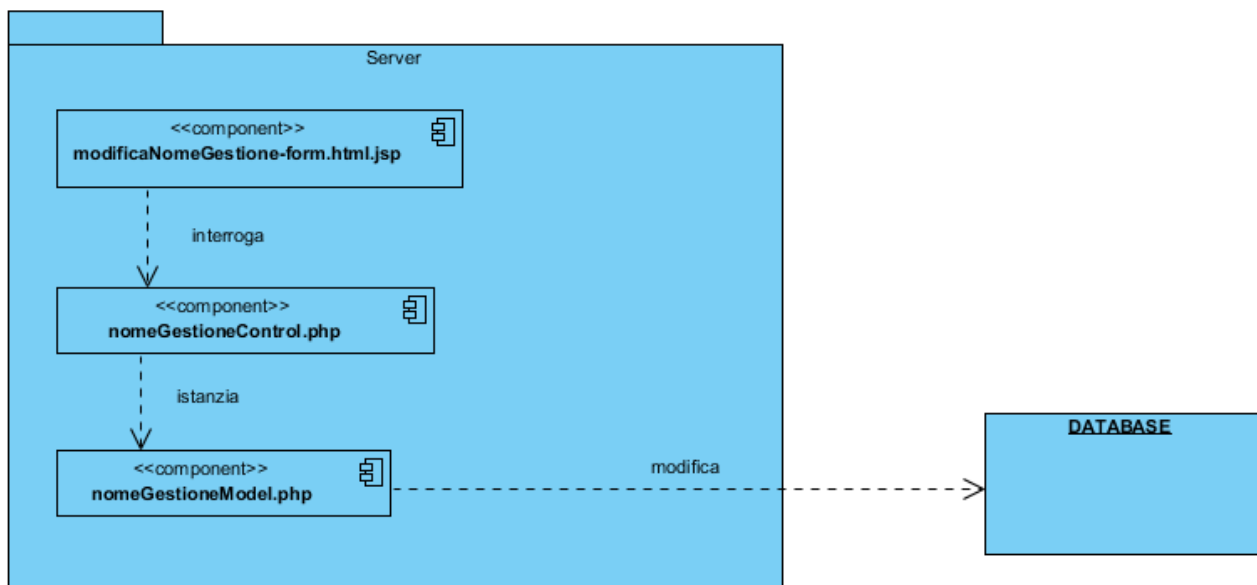
Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole visualizzare una lista di entità pre-esistenti.

Aggiunta



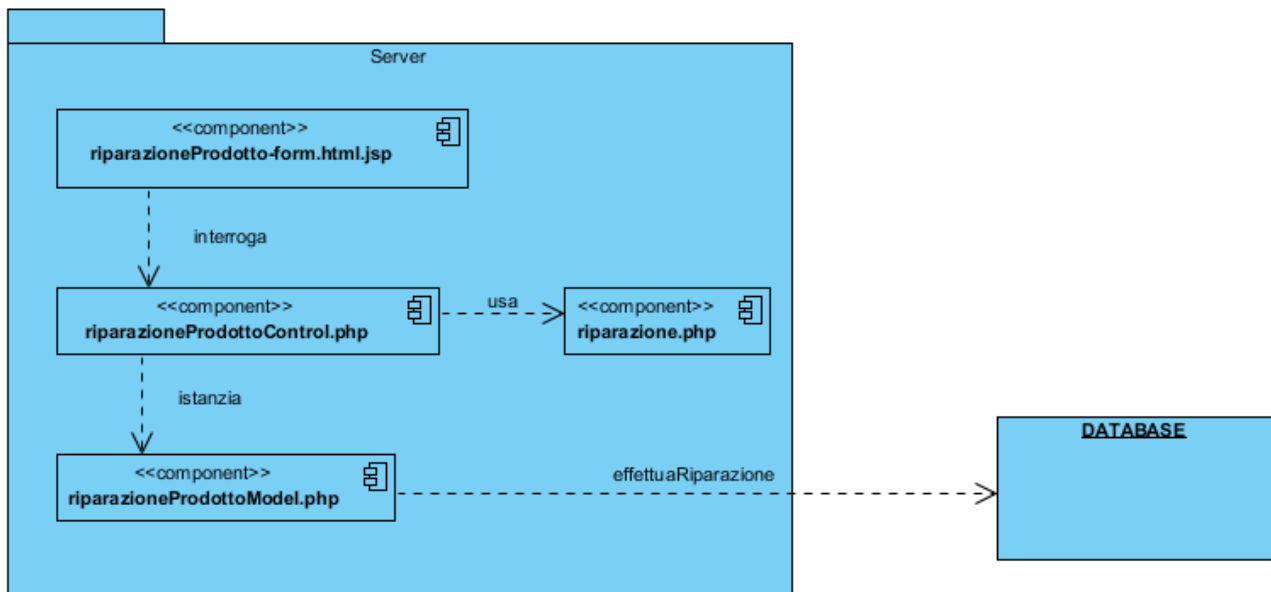
Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole effettuare un'aggiunta.

Modifica



Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole effettuare una modifica su un'entità già esistente.

Riparazione

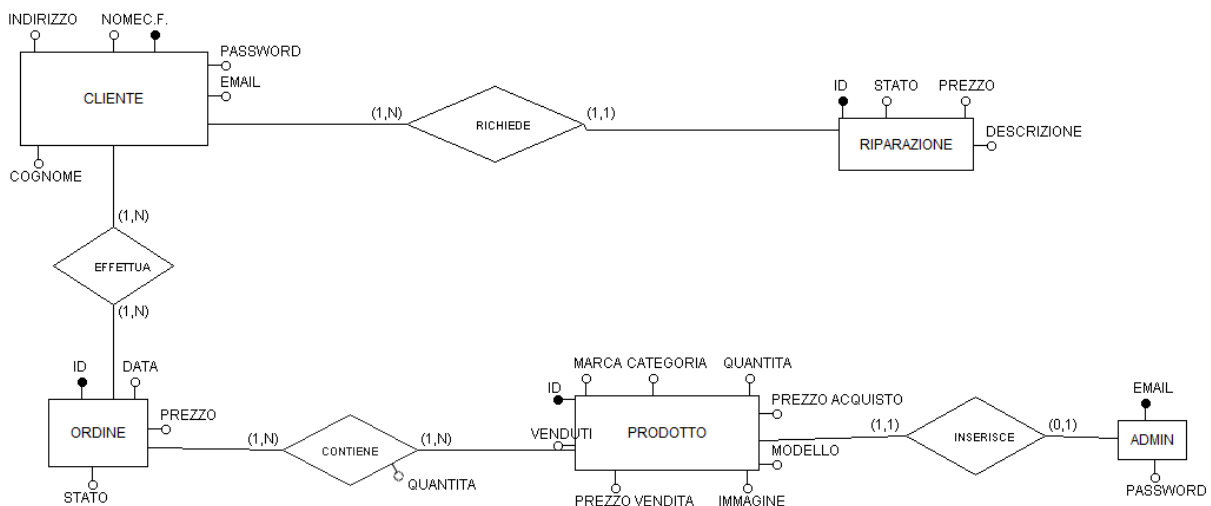


Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole effettuare una riparazione.

Gestione dei dati persistenti

La scelta dei dati persistenti è stata fatta analizzando tutti gli oggetti rilevanti che in caso di chiusura controllata/incontrollata dovranno essere conservati.

Schema EER



Dizionario dei dati

Relazioni

Contiene:

Si usa la tabella acquisto, che associa all'ordine i prodotti acquistati

Chiavi Esterne	<i>Relazione tra venditore e ordine</i>
<u>Codice prodotto</u>	Chiave esterna codice fiscale del cliente
Quantità	Quantità del prodotto acquistato
<u>Codice ordine</u>	Chiave esterna codice dell'ordine

Effettua:

Chiavi Esterne	<i>Relazione tra venditore e ordine</i>
<u>Codice Fiscale</u>	Chiave esterna codice fiscale del cliente
<u>Codice ordine</u>	Chiave esterna codice dell'ordine

Richiede:

Chiavi Esterne	<i>Relazione tra venditore e ordine</i>
<u>Codice Fiscale</u>	Chiave esterna codice fiscale del cliente
<u>Codice Riparazione</u>	Chiave esterna codice della riparazione

Ordine:

Identifica un ordine.

Campo	Ruolo
IdO	Contiene Codice alfanumerico che identifica l'ordine
Data	Contiene la data in cui è stato effettuato l'ordine
Prezzo	Contiene il prezzo dell'ordine
Stato	Contiene lo stato dell'ordine

Amministratore:

Identifica i dati dell'amministratore all'interno del database.

Campo	Ruolo
Email	Contiene l'indirizzo email dell'amministratore.
Password	Contiene password del dell'amministratore

Cliente:

Identifica un cliente all'interno del database.

<i>Campo</i>	<i>Ruolo</i>
C.F	Codice alfanumerico che identifica il cliente
Telefono	Contiene Numero telefonico del cliente
Nome	Contiene Nome del cliente
Cognome	Contiene Cognome del cliente
Email	Contiene l'indirizzo email del cliente
Password	Contiene password del cliente
Indirizzo	Contiene l'indirizzo del cliente

Riparazione:

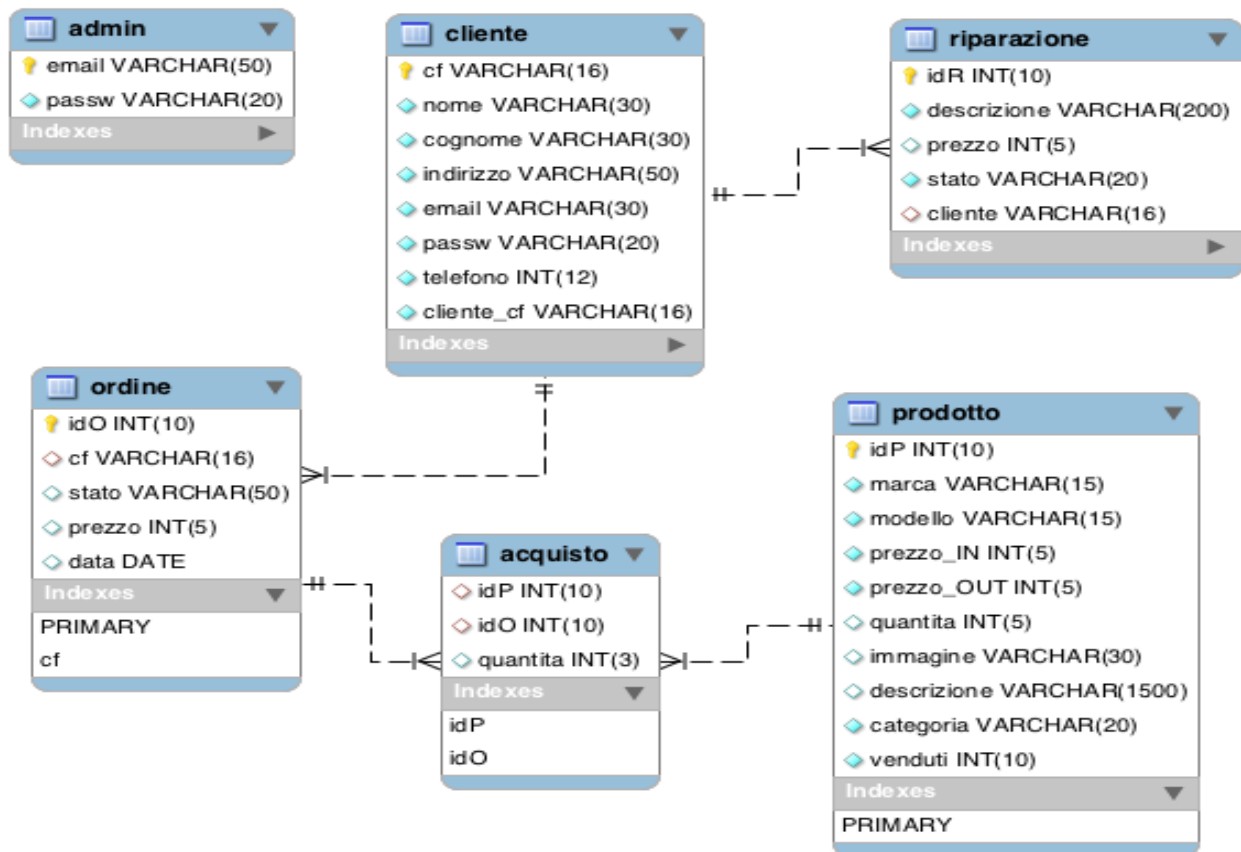
Identifica una riparazione all'interno del database.

<i>Campo</i>	<i>Ruolo</i>
IdR	Contiene il Codice che identifica la riparazione.
Prezzo	Contiene il prezzo della riparazione
Descrizione	Contiene la descrizione del problema da risolvere
Stato	Contiene lo stato della riparazione "ACCETTAZIONE" "RICHIESTA" "RIFIUTATO" "ACCETTATO" "RIPARAZIONE IN CORSO" "RIPARATO"

Prodotto:

Identifica un prodotto all'interno del database.

<i>Campo</i>	<i>Ruolo</i>
IdP	Codice alfanumerico che identifica univocamente il prodotto
Marca	Contiene la marca del prodotto
Modello	Contiene il modello specifico del prodotto
Prezzo di acquisto	Contiene il prezzo di acquisto del prodotto
Prezzo di vendita	Contiene il prezzo di vendita del prodotto
Venduti	Contiene il numero di oggetti venduti di quel prodotto
Immagine	Contiene il nome del file dell'immagine del prodotto
Categoria	Contiene la categoria del prodotto
Descrizione	Contiene la descrizione del prodotto
Quantità	Contiene la quantità del prodotto



Codice di creazione del database

```
drop database if exists negozio;
create database StaySoftwareDB;
use StaySoftwareDB;
```

```
create table admin
( email varchar(50) not null,
  passw varchar(20) not null
);
```

```
create table cliente
( cf varchar(16) primary key,
  nome varchar(10) not null,
  cognome varchar(10) not null,
  indirizzo varchar(20) not null,
  email varchar(50) not null,
  passw varchar(20) not null,
  telefono int(10) not null
);
```

```
CREATE TABLE prodotto
( idP int(10) PRIMARY key AUTO_INCREMENT,
  marca varchar(10) not null,
  modello varchar(15) not null,
```

```
prezzo_IN int(5) not null,  
prezzo_OUT int(5) not null,  
quantita int(5),  
immagine varchar(15),  
categoria varchar(10),  
descrizione varchar(200)  
);
```

```
create table riparazione  
( idR int(10) PRIMARY key AUTO_INCREMENT,  
  descrizione varchar(200) not null,  
  prezzo int(5),  
  stato varchar(20) not null,  
  cliente varchar(16),  
  foreign key(cliente) references cliente(cf)  
);
```

```
create table ordine
```

```
(  
idO int(10) PRIMARY key AUTO_INCREMENT,  
cf varchar(16),  
  stato varchar(50),  
  prezzo int(5),  
  data date,  
  foreign key(cf) references cliente(cf)  
idP int(10) ,  
quantita int(3),  
  
  foreign key(idP) references prodotto(idP),  
);
```

Controllo degli accessi e sicurezza

Controllo degli accessi

Il sistema prevede due tipi di attori:

- **Amministratore:** è l'amministratore del sistema ed ha accesso a tutte le componenti. È unico e può inserire nuovi prodotti, modificare la quantità, il prezzo e gestire le riparazioni
- **Cliente:** è colui che si registra sulla piattaforma e usufruisce dei servizi che essa propone. Può acquistare beni e/o servizi, richiedere una riparazione, accettare il preventivo della riparazione.

Gli oggetti che vengono gestiti direttamente dagli attori sono:

- *Ordine*
- *Cliente*
- *Riparazione*
- *Prodotto*

<i>Attore/Oggetto</i>	<i>Amministratore</i>	<i>Utente</i>
<i>Ordine</i>	- - V -	C - V -
<i>Cliente</i>	- - V -	C M V -
<i>Riparazione</i>	- M V -	C M V -
<i>Prodotto</i>	C M V E	- - V -

C *Crea*
M *Modifica*
V *Visualizza*
E *Elimina*

Sicurezza

Il software assicura la protezione dei dati tramite un controllo degli accessi al sistema.

Ogni utente può accedervi attraverso l'username e la password che è stata scelta in fase di registrazione e in nessun altro modo.

Ogni accesso al sistema sarà registrato in un file di log che permetterà all'amministratore di avere sempre il controllo sulla situazione degli accessi (file di log del server).

Il software, inoltre, garantisce che non sia possibile accedere all'area utente e/o acquistare prodotti senza che l'utente sia registrato.

Nel caso vengano inseriti dati sbagliati durante la fase di accesso alla piattaforma, il sistema deve avvertire l'utente dell'errore commesso e mostrare nuovamente la schermata di accesso.

La sicurezza del profilo utente è fortemente legata alla scelta della password (una password che non richiama parole di uso comune, che ha al suo interno lettere maiuscole e minuscole e che è caratterizzata dalla presenza di caratteri numerici sarà più sicura di un nome proprio o di un nome che richiama il linguaggio di uso comune).

I canoni che username e Password devono rispettare sono i seguenti:

- *Username*: coincide con la mail, questo garantisce ulteriormente l'univocità di quest'ultima.
- *Password*: deve avere una lunghezza compresa fra un minimo di 7 ed un massimo di 20 caratteri, non può contenere caratteri.

Gestione del controllo globale

Il flusso di controllo globale del sistema software in questione è gestito da un **“Event-driven control”**.

Con questo tipo di controllo il flusso del programma è largamente determinato dal verificarsi di eventi esterni infatti il flusso di eventi principale una volta ricevuto un input lo trasferisce al relativo oggetto che manda in esecuzione un sottosistema.

È stata fatta questa determinata scelta perché tale meccanismo di flusso è particolarmente flessibile ed adatto alle interfacce utenti e soprattutto sfrutta appieno le potenzialità del linguaggio di programmazione usato per sviluppare questo sistema.

Boundary conditions

Configuration

Di seguito sono riportate le fasi in cui gli oggetti vengono creati, resi persistenti o distrutti.

- *Ordine*
 - *Viene creato ogni volta che un cliente procede all'acquisto di beni e/o servizi;*
 - *Sono memorizzati sul database e vengono associati ai clienti.*
- *Amministratore*
 - *Viene creato con il rilascio del sito, può essere cambiato solo accedendo al database, nel caso in cui l'organico aziendale viene ampliato l'amministratore inserisce i vari Amministratore del sito sempre accedendo direttamente al database;*
 - *È memorizzato nel database.*

- *Cliente*
 - *Viene creato nel momento in cui qualcuno effettua la registrazione sulla piattaforma;*
 - *È memorizzato in modo permanente sul database e identifica univocamente un cliente collegandolo a tutti gli ordini che ha effettuato.*
- *Riparazione*
 - *Viene creata dal cliente nel momento in cui è richiesta;*
 - *È memorizzata nel database in modo permanente ed è collegata al cliente che l'ha richiesta.*
- *Prodotto*
 - *Viene creato dall'Amministratore e poi aggiornato in termini di prezzi e quantità;*
 - *È memorizzato nel database in modo permanente e viene costantemente aggiornato in base alle variazioni della quantità o a eventuali aumenti/diminuzioni di prezzo;*
- *Acquisto*
 - *Viene creato durante la creazione dell'ordine, per ogni modello di prodotto inserito nell'ordine*
- *Carrello*
 - *Viene creato nel momento in cui il cliente inizia ad acquistare prodotti*
 - *Viene eliminato nel momento in cui il cliente completa l'ordine oppure si disconnette*

Exception handling

- *Connessione al database fallita:* Errore del sistema durante la connessione al database, questo comporta l'impossibilità di accedere ai dati e di inserire dati. In questo caso, il sistema, riprova la connessione e nel caso in cui non riesce a connettersi al database comunicherà l'errore all'Amministratore e tenterà la connessione ad un database di backup.
- *Interruzione inaspettata del sistema:* essa può accadere per vari motivi come ad esempio un blackout elettrico (gestito attraverso l'installazione di un gruppo di continuità) oppure per un errore di sistema che lo costringe al riavvio. Questo tipo di problematiche sono gestite singolarmente.

Nome scenario	SC_Startup Server
Istanze di attori partecipanti	Marco: Amministratore
Flusso di Eventi Amministratore	<p>1. Marco, amministratore di StaySoftware, vuole avviare il sistema e quindi clicca sul pulsante "Avvia".</p> <p>2. Il sistema attiva i server e i relativi servizi in remoto, con le opportune procedure di avvio.</p> <p>3. Il sistema notifica il successo dell'operazione.</p>

Nome scenario	SC_Shutdown Server
Istanze di attori partecipanti	Marco: Amministratore
Flusso di Eventi Amministratore	<ol style="list-style-type: none"> 1. Marco, amministratore di StaySoftware, vuole arrestare il sistema e quindi preme sul pulsante "Arresta". 2. Il sistema controlla eventuali Client ancora connessi, e ne termina la connessione. 3. Tramite le opportune procedure, il sistema disattiva i servizi in remoto e il server. 4. Il sistema notifica il successo dell'operazione.

ID	UC_Startup
Nome use case	Startup Server
Partecipanti	Amministratore
Condizioni di ingresso	L'amministratore accede al sistema
Flusso di eventi	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema e clicca sul pulsante "Avvia". 2. Il sistema attiva i propri servizi rendendosi disponibile in tutte le sue funzionalità e notifica il successo dell'operazione
Eccezioni	Errore Startup
Condizioni di uscita	Il sistema è attivo con tutte le sue funzionalità

ID	UC_Shutdown
Nome use case	Shutdown Server
Partecipanti	Amministratore
Condizioni di ingresso	L'amministratore accede al sistema
Flusso di eventi	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema e clicca sul pulsante "Spegni". 2. Il sistema effettua una scansione per verificare client connessi e li

	disconnette; avvia la procedura di arresto. Il sistema notifica il successo dell'operazione
Eccezioni	Errore Shutdown
Condizioni di uscita	Il sistema viene arrestato

Glossario

Termine	Descrizione
<i>Sistema</i>	Si intende il sistema che verrà sviluppato.
<i>Architettura</i>	Struttura del sistema.
<i>Attori</i>	Gli utenti tipo che utilizzeranno il sistema.
<i>Affidabilità</i>	L'attitudine del sistema a svolgere la funzione richiesta in determinate condizioni per un dato intervallo di tempo.
<i>Estensibilità</i>	Capacità di estensione del sistema.
<i>Client</i>	Chi accede ai componenti e ai servizi di un server.
<i>Server</i>	Chi offre componenti e servizi.
<i>Boundary conditions</i>	Condizioni di uscita/errore/inizializzazione/chiusura
<i>Event-driven control</i>	Controllore di eventi
<i>Responsive</i>	Che si adatta alle dimensioni del dispositivo
<i>Utente non loggato</i>	Il termine identifica le persone che non sono loggate al sistema, non possono fare acquisti né richiedere riparazioni
<i>Utente loggato</i>	Il termine identifica le persone che sono correttamente loggate al sistema
<i>Carrello</i>	L'insieme di prodotti che il cliente vuole acquistare
<i>Ordine</i>	L'insieme di prodotti acquistati in una sessione dal cliente
<i>Riparazione</i>	Servizio offerto dal negozio, che viene richiesto dal cliente. Di solito la riparazione di un prodotto.
<i>Preventivo</i>	Il prezzo della riparazione inserito dall'amministratore in base al quale il cliente decide se accettare o meno il costo della riparazione
<i>Amministratore</i>	Il termine identifica la persona che gestisce la piattaforma